RESEARCH ARTICLE | MAY 18 2020

Is it necessary to resolve the Debye length in standard or δf PIC codes?

B. F. McMillan

Phys. Plasmas 27, 052106 (2020)
https://doi.org/10.1063/1.5139957

Physics of Plasmas ARTICLE scitation.org/journal/php

Is it necessary to resolve the Debye length in standard or δf PIC codes?

Cite as: Phys. Plasmas **27**, 052106 (2020); doi: 10.1063/1.5139957
Submitted: 18 December 2019 · Accepted: 23 April 2020 ·
Published Online: 18 May 2020

B. F. McMillan[a)]

AFFILIATIONS

Centre for Fusion, Space and Astrophysics, Department of Physics, University of Warwick, CV4 7AL Coventry, United Kingdom

[a)]Author to whom correspondence should be addressed: B.F.McMillan@warwick.ac.uk

ABSTRACT

Numerical instabilities of standard Particle-In-Cell (PIC) codes were observed and explained very early in their development. In the zero-time step limit, these instabilities arise from the interaction between the spatial grid and the (artificial) particle *shape functions*. δf PIC codes, which have recently been especially popular in gyrokinetic simulations, suffer from similar instabilities. In the zero time step limit, the numerical stabilities of standard and δf methods are equivalent. Numerical instabilities arise when the simulation grid does not "resolve the Debye length," but many modern PIC codes use relatively high order shape functions, and as a result, the worst-case numerical growth rates are undetectably small; in addition, some codes use energy-conserving methods which usually prevent this numerical instability from arising. Similarly, a numerical instability was found in a gyrokinetic δf code using a first-order shape function; we show that this is related to the usual PIC numerical instability. In the gyrokinetic case, where waves have acoustic dispersion at a large wavenumber, increasing the grid resolution actually increases the growth rate of the numerical instability, and the prescription of resolving an effective Debye length is not applicable. However, using higher order shape functions is still an effective remedy.

I. INTRODUCTION

The stability of Particle-In-Cell (PIC) codes has been explored for historical schemes of various orders,[1] for a δf scheme,[2] and with modified current deposition schemes.[3] Sufficiently small time steps and spatial grid spacing, as well as a sufficient number of computational markers, are required for stable solutions of PIC algorithms. It is often stated, for example, that PIC methods must resolve the Debye length for stability, but in practice, modern PIC codes with high order splines are often run with grids that very strongly under-resolve the Debye length without obvious instability or numerical self-heating. In the less well-explored case of gyrokinetic PIC simulation, the same numerical issue arises:[4] we explain the correspondence between numerical stability in the usual Vlasov–PIC method and in gyrokinetic-PIC simulations. As a first step, we will recap some of the stability theory of PIC methods in the large-marker number limit. We then demonstrate by direct numerical solution that standard and δf PIC simulations are subject to instabilities predicted by theoretical analysis of the numerical scheme. This explains why a numerical instability was found in the study by Wilkie and Dorland:[4] the instability is in fact the usual numerical instability associated with the PIC method, but in unfamiliar context.

II. THE GYROKINETIC AND STANDARD PIC PROBLEM

To explain the commonality between the different problems, we translate the gyrokinetic problem into a form close to the 1D Vlasov–Poisson system. For simplicity, we will treat the perpendicular direction spectrally. The linearized continuum equations (4) and (5) in Ref. 4 may be rewritten in a similar form to the conventional 1D Vlasov–Poisson system, as

$$\frac{\partial f}{\partial t} + v_{||}\frac{\partial f}{\partial x} = \frac{e}{T_e} v_{||} \frac{\partial F_0}{\partial v}\frac{\partial \phi}{\partial x}, \tag{1}$$

for the electron perturbed distribution function f and background distribution function F_0 (the ion parallel motion is slow and may be ignored), and

$$\frac{n_i k_\perp^2 \rho_i^2}{2 T_i S_p^2}\phi = -\int_{-\infty}^{\infty} f dv_{||}. \tag{2}$$

Here, the perpendicular direction has been treated spectrally for simplicity, and the factor S_p^2 has been introduced to account for the use of shape functions in the direction perpendicular to the field and we take $S_p^2 = 1$ (this is appropriate when the perpendicular wavelength is long compared to the grid spacing, which is the most problematic case).

The notation can be simplified by normalizing the velocity to $v_{te} = (T_e/m_e)^{1/2}$, normalizing the potential to $\phi_0 = e/T_e$, time by $1/v_{te}$, and scaling the distribution function by the background density n, to give

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = v\frac{\partial F_0}{\partial v}\frac{\partial \phi}{\partial x} \tag{3}$$

and

$$\phi = -\zeta_0^2 \int_{-\infty}^{\infty} f dv, \tag{4}$$

with $\zeta_0 = 1/k_\perp \rho_i$ (note that x is not normalized). Interestingly, there is no characteristic parallel length scale or timescale in this problem. In the cold-plasma limit, solutions to this problem are acoustic waves with dispersion $\omega^2 = \zeta_0^2 k^2$. This system of equations is equivalent to the problem studied in Ref. 2, of ion-acoustic waves (with PIC ions and adiabatic electrons), where the electric potential is proportional to the (numerical) ion density perturbation, but with ions and electrons swapping roles.

These problems are only slightly different to the usual electron plasma wave problem (considered by Ref. 1) where for velocity in units of v_{te} and x in units of the Debye length, the equation for the (non-dimensionalized) field ϕ is replaced by

$$\frac{\partial^2 \phi}{\partial x^2} = \int_{-\infty}^{\infty} f dv. \tag{5}$$

For this problem, a characteristic length scale (the Debye length) and timescale (plasma wave inverse frequency) appear.

III. THE PIC METHOD IN THE LARGE-MARKER NUMBER LIMIT

A. Standard PIC

The standard PIC method uses a set of M markers at positions x_i and velocities v_i with weights w_i. We can define a simulation distribution f via

$$\int_C f dx dv = \sum_i \int_C w_i \delta(x - x_i)\delta(v - v_i) dx dv, \tag{6}$$

in the large-M limit for any region C of phase space. At this point, this simply serves as a definition of the distribution f, but this can be shown to satisfy a modified Vlasov equation, and with an appropriate initial condition, thereby model the time-evolution of a plasma. We will analyze the PIC method for a single species to simplify the notation, assuming that a neutralizing background is present to ensure quasi-neutrality of the unperturbed plasma.

In the standard PIC method, the electric field $\bar{E}_j$ (i.e., used for calculating marker acceleration) at position x is evaluated using

$$\bar{E}(x) = \sum_j E(x_j) S(x - x_j), \tag{7}$$

with $E(x_j)$ the electric field stored at evenly spaced grid positions $x_j = j\delta x$, and S a "Shape function," which is a narrow compact support function, with the property $\sum S(x_j - x) = 1$ for all x. Usually S is monotonically increasing to some peak value at $x = 0$, and symmetric about $x = 0$.

"Energy-conserving" PIC methods use

$$\bar{E}(x) = -\sum_j \phi(x_j) \frac{\partial}{\partial x} S(x - x_j), \tag{8}$$

but will not be discussed in detail: note that the numerical electric field is exactly minus 1 times the spatial derivative of the potential. The principle of such schemes is that the numerical method is derived based on a system Lagrangian,[5] and this directly implies conservation of energy due to time-symmetry (but not momentum conservation because the grid removes the continuous translation symmetry). The energy-conservation property allows one to show that they will not be subject to the numerical instability under discussion[6] if the background plasma is stationary: significant instability is still however possible for a plasma with drift comparable to the thermal speed. Note that energy-conserving methods have been used effectively in the context of gyrokinetic simulation for some time.[7]

The charge associated with each grid point ("charge deposition") is then defined using

$$n_j = \frac{1}{\delta x} \sum_i S(x_j - x_i) w_i, \tag{9}$$

and the Poisson equation at point j may be defined using

$$\epsilon_0 \left[\frac{\partial}{\partial x} E\right]_j = q n_j. \tag{10}$$

Some kind of numerical approximation is required to define the spatial derivative on the left-hand side, and a simple one is to use a spectral treatment $\partial/\partial x = ik$, which is what we do here. Other choices will lead to an effective (wavelength-dependent) change to the coupling coefficient ϵ_0 by a factor of order unity.

In the large marker-number, zero-time step limit, standard kinetic theory can be used to find the numerical Vlasov equation, for the distribution of markers g in the simulation, that represents the number of markers per unit phase space volume. The usual Vlasov equation with a modified electric field

$$\frac{\partial g}{\partial t} + v\frac{\partial g}{\partial x} + \frac{q\bar{E}}{m}\frac{\partial g}{\partial v} = 0, \tag{11}$$

is recovered in the nonrelativistic limit. For general weights w_i, we have $f = g\langle w_i \rangle$ (the average is over small regions of phase space in the large marker-number limit), but in the usual case where marker weights are constant we have $N_0 g = Mf$, where $N_0 = \int n dx$ is the total number of plasma particles. We rewrite, and then solve, these equations in terms of the more familiar plasma distribution function f in the remainder of the paper. Because of the PIC grid, this equation only has a discrete translational symmetry, rather than a continuous one.

The charge density becomes

$$n_j = \frac{1}{\delta x} \int S(x - x_j) f(x, v) dx dv, \tag{12}$$

and this is used to evaluate the electrostatic potential using the Poisson equation.

B. δf PIC

In the δf method, we have f (the numerical distribution function) defined as

$$\int_C f dx dv = \int_C f_0 dx dv + \sum_i \int_C w_i \delta(x - x_i)\delta(v - v_i) dx dv, \quad (13)$$

on any region C, where f_0 may have general time and space dependence, but will be chosen to be a spatially and temporally uniform Maxwellian here. We define $\delta f = f - f_0$. The markers are taken to nominally represent a nearby phase space of fixed volume Ω_i, which is initialized to $\Omega_i = 1/g$ when markers are loaded, and for marker loading proportional to the background distribution $\Omega_i = N_0/Mf_0$. The weight evolution equation is

$$\frac{dw_i}{dt} = -\Omega_i \frac{df_0}{dt_i}, \quad (14)$$

where the convective derivative is associated with the trajectory of marker i.

As has been previously stated,[2] the numerical Vlasov equation (found in the zero-time step limit) for standard and delta-f PIC is equivalent; a derivation of this result is given in Appendix A. In short, all that is required is for the background distribution evaluated using the markers to cancel the analytical evaluation. Both methods satisfy a modified Vlasov equation, and at wavelengths long compared to the spatial grid, the modified numerical Vlasov equation is equivalent to the electrostatic Vlasov equation of physical interest.

IV. STABILITY OF THE PIC METHOD

We now consider the analytical theory for the stability of PIC methods in the large-marker number and zero-time step limit. This is approached by linearizing the numerical Vlasov equation, about an initially spatially homogeneous state with $E = 0$. As usual, we take a Maxwellian background distribution $f_0 \propto \exp(-mv^2/2T)$. A plane wave disturbance with wavelength k of the electric field is considered so $E_j = \tilde{E}\exp(ix_j k)$. The numerical Vlasov equation may then be solved, with the splitting $f = f_0 + f_1$ and linearizing to find an equation for f_1. We are working with f rather than g since the distinction is not relevant in the infinite-marker number limit.

Note that due to the shape functions, the markers are not subject to a plane wave electric field, but (normally) a piecewise-polynomial approximation to a plane wave; the Fourier transform of this piecewise polynomial involves not just the wavenumber k but also "alias waves" at periods $k + n2\pi/\delta x$, for integer n. This resulting distribution function f_1 is then inserted into the numerical Poisson equation, and the normal modes of the numerical system are then found. That is, conceptually, what we are doing for the numerical system is equivalent to solving the warm-plasma dispersion relation of a real plasma.

This analysis is found in classic texts[1] as well as more recent papers and we don't repeat it in detail. Reference 2 gives analytical expressions for the dielectric response ϵ of a Particle-In-Cell numerical plasma, which gives the (Fourier coefficient) of the density response for a given potential field, and is defined using

$$\int dv f = \delta n = \epsilon(k, \omega)\phi. \quad (15)$$

We will specialize to the zero-time step limit, in which case, both Eqs. (53) (for the δf dielectric response) and (55) (for the full-f dielectric response) of Ref. 2 simplify to

$$\epsilon(k, \omega) = -k \sum_q S(k_q)^2 \int dv \frac{\partial f_0}{\partial v} \frac{1}{\omega - k_q v}, \quad (16)$$

the sum over all integers q, $k_q = k + 2\pi q/\delta x$, and

$$S(k) = \left(\frac{\sin(k\delta x/2)}{k\delta x/2}\right)^{N+1}. \quad (17)$$

Here, δx is the grid spacing and N is the order of the particle shape spline function. The existence of a single numerical dispersion relation is sufficient to show that full-f and δf methods are numerically equivalent in terms of stability in the zero-time step limit. $\epsilon(k, \omega)$ differs from the continuum dielectric response due to the inclusion of "alias" waves at wavenumber k_q, which occur because the field the particles see is not an exact sine wave, but a piecewise polynomial approximation to a sine; this can be decomposed into a set of Fourier modes indexed by q. The resonance of particles with these alias wave velocities can be seen as the cause of unphysical mode growth or damping in the PIC method.

Reference 2 uses these expressions for ϵ to solve the numerical dispersion relation for a specific ion-acoustic model, and find that stability decreases as the ratio ζ_0 of the ion acoustic wave speed to the ion thermal speed increases: for the range of parameters explored in that reference, the numerical plasma was always stable at zero time step, but if they had explored a slightly higher ζ_0 they would have seen a zero-time step numerical instability (note however that the growth rate peaks at finite ζ_0 and asymptotes to zero for $\zeta_0 \to \infty$, so the cold-ion limit is stable as expected). The issue (both for ion acoustic waves and electron plasma waves) is that waves propagating much faster than ion thermal velocities are not subject to physical damping, so small amounts of unphysical drive (due to numerical aliasing) result in a numerical instability. In the gyrokinetic case, there will normally be ω_H modes in the system that propagates much faster than the electron thermal velocity, which allow this numerical instability.

For the 1D Vlasov–Poisson equation with a neutralizing fixed ion density, the numerical dispersion relation in normalized units can be written, using the Poisson equation, $-k^2\phi = \delta n$, and Eq. (16) as

$$0 = \Omega_{VP}(k, \zeta, k\delta x) = 1 + \frac{1}{k^2} \sum_q \left(\frac{\sin(k\delta x/2)}{k\delta x/2 + q\pi}\right)^{2N+2} \times \int dv \frac{\partial f_0}{\partial v} \frac{1}{\zeta - v[1 + 2\pi q/k\delta x]}, \quad (18)$$

where $\zeta = \omega/k$ is the numerical wave speed. For the gyrokinetic and ion-acoustic problems, where we have $\phi = -\zeta_0^2 \delta n$, the numerical dispersion relation is found using Eq. (16) as

$$0 = \Omega_{GK}(k, \zeta, k\delta x) = 1 + \zeta_0^2 \sum_q \left(\frac{\sin(k\delta x/2)}{k\delta x/2 + q\pi}\right)^{2N+2} \times \int dv \frac{\partial f_0}{\partial v} \frac{1}{\zeta - v[1 + 2\pi q/k\delta x]}. \quad (19)$$

The numerical gyrokinetic (GK)/ion-acoustic relation has the somewhat special property of being a function of wave phase speed ζ and grid-normalized wavenumber ($k\delta x$) only. This is a consequence of the absence of a characteristic spatial scale in the GK/ion-acoustic problem, so only the grid scale appears.

These dispersion relations only differ in the prefactor multiplying the sum. Thus, given a solution $\Omega_{GK}(k,\zeta,k\delta x)=0$ for some ζ_0 we have $\Omega_{VP}(1/\zeta_0,\zeta,k\delta x)=0$. Note that this explains the fact[15] that the numerical instability growth rate in the GK simulations is inversely proportional to the grid spacing: solutions of this dispersion relation for the most unstable value of $k\delta x$, have fixed $\zeta=\omega/k$ and the growth rate $\mathrm{Im}(\omega)$ therefore scales like $1/\delta x$. This contrasts with the usual situation in 1D Vlasov–Poisson simulations, where the unphysical instabilities may be suppressed by reducing δx (*resolving the Debye length*).

These dispersion relations may be directly numerically evaluated (for growing modes with $\mathrm{Im}(\omega)>0$) to find any numerical instabilities, and we will do this later for a particular case. Approximate analytical expressions may also be found for waves with $\zeta\gg 1$ and under the assumption that $S(k_q)\ll S(k)$ for $q\neq 0$; the real part of the dispersion relation is then dominated by the cold-ion response (in the $q=0$ term) and the growth rates are found as a correction due to the other terms in the sum (mostly $q=\pm 1$), in the small growth rate limit. This identifies the growing waves as standard plasma waves, modified slightly by numerical aliasing, and with frequencies reduced slightly due to smearing out of the particle-field interaction by the shape function. For higher order shape functions, the $q=0$ term becomes even more dominant and growth rates of this numerical instability rapidly decrease.

V. NUMERICAL RESULTS FOR CONVERGED AND NON-CONVERGED SIMULATIONS

To demonstrate the numerical instability in PIC simulations, and how it may be avoided, we run 1D Vlasov–PIC simulations (in the non-relativistic, electrostatic limit) using the EPOCH code,[10] which has recently been extended to allow δf simulations. The code defaults to using third-order shape functions, partly to avoid exactly this kind of instability (this appears to be a standard practice for modern PIC simulations in laser-plasma applications) but can be set to use first-order splines for testing purposes. EPOCH uses a leapfrog-time time-integration and an Esirkepov-type current deposition scheme[11] (this is an extension of the Villasenor–Buneman[12] approach to higher-order shaper functions) and is thus somewhat different to historical PIC algorithms. We also wrote a very simple PIC code to check that these results are insensitive to using the historical methods of Ref. 1 instead of Esirkepov current deposition: the full-f simulation was run using this code. A time step $t\omega_{pe}=0.1$ was used.

We run full-f and δf cases, with both first and third order splines, for the nominal parameters of Ref. 4, with number of grid points $N_x=4$ and $\delta x=10\lambda_{de}$. The full-f case is run with 10^7 markers and otherwise, 10^5 markers are used. We also run simulations with $N_x=40$, but the same simulation length, so the grid spacing is ten times smaller, and $\delta x=\lambda_{de}$. Plasma waves are initialized using a sinusoidal velocity perturbation of magnitude $10^{-4}v_{th}$, where v_{th} is the thermal velocity. Physically, the linear damping mechanism is a resonance with particles traveling at the wave phase speed, which is $\sim 10v_{th}$: the Maxwellian distribution function is essentially zero there so the correct damping rate is effectively zero. We plot the time-evolution of the mean squared electric field (Fig. 1), which oscillates with twice the plasma frequency as we have initialized a standing wave.

For both full-f and δf, a numerical instability for the mode $k\delta x=\pi/2$ with a growth rate of $0.01\omega_{pe}$ is found for the first-order splines but with third-order splines, the wave has a small positive damping rate of $0.002\omega_{pe}$. The simulation with first order splines but increased resolution has a very small positive damping rate of $5\times 10^{-4}\omega_{pe}$. The frequencies measured in the PIC simulations are all somewhat different to the correct wave frequency which is to be expected since the wave is mostly poorly resolved by the grid.

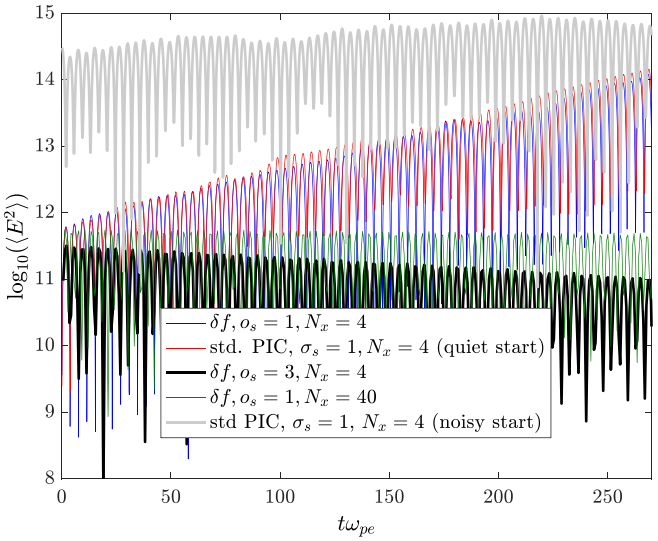

FIG. 1. Mean squared electric field amplitude vs time for a plasma oscillation in simulations of a 1D Vlasov–Poisson system. The standard PIC simulation is labeled with a red curve. The δf simulations have blue and black traces, using first and third order splines, respectively. The spline order is denoted o_s. A simulation with first order splines and $N_x=40$ so that the grid resolves the Debye length is also shown (green trace).

The full-f simulations have nearly identical behavior to the δf simulations, demonstrating that the instability here and in Ref. 4 is not a particularity of the δf method. Normally numerical self-heating[1,10] is observed in PIC simulations rather than the numerical instability predicted from linear stability analysis, but only two modes are present in this simulation, and we are using an unusually large number of markers per cell, four orders of magnitude larger than in typical PIC simulations. In general, the noise in a conventional PIC simulation would hide these fairly subtle numerical instabilities, in addition to potentially hiding actual physical instabilities that are of interest, but for this very small test case, the instability is clear. To see this instability cleanly in standard PIC, even with millions of markers per grid cell, a quiet-start is required: little growth of the mode is seen in the "noisy start" simulation, as the instability saturation level is not much higher than the initial noise level.

Translated into the units of Ref. 4, the growth rate for the $N_x=4$ case with first order shape functions is $1.0c_s/a$, which is about a factor of two lower than the growth rate found there: the reason for the difference is unclear, but some difference might be expected due to the 1D rather than 2D discretization (or equivalently, the spectral treatment of the perpendicular direction).

A comparison between PIC results and the analytical formula [Eq. (18)] for instability is also made for the case $\delta x=10\lambda_{de}$, with first order splines, and 100 simulation grid points. We examine the wavenumber dependence of the instability by isolating a single Fourier component of the potential, setting all other modes in the simulation

to zero at each time step. The agreement between the analytical theory and the PIC result curve in Fig. 2 is generally excellent. We also plot the analytically predicted growth rate of the numerical instability for a PIC simulation with third order splines (scaled by a factor of 100): these growth rates are close enough to zero that they are negligible for most practical purposes.

Note that Ref. 4 also find that using splines for charge assignment of one order higher than for the field interpolation removes the instability (in the case of Ref. 4, method C, using piecewise-constant fields): this is a so-called energy-conserving method[1] known to have favorable stability properties for a cases near an un-shifted Maxwellian. Energy-conserving methods are thus a natural choice for gyrokinetic PIC simulation[7] where the plasma is near-stationary and near-Maxwellian. Various other methods exist to avoid this numerical instability, but these have certain limitations. The solution favored by modern PIC simulations is generally to either resolve the (effective) Debye length or use higher-order splines. We have shown that both of these almost completely suppress the numerical instability for Vlasov–Poisson. Higher order splines will also suppress the instability in the gyrokinetic formalism. On the other hand, higher resolution in the parallel direction actually increases the numerical instability growth rate for gyrokinetic simulations: resolving the Debye length translates into a condition that is not possible to satisfy in the gyrokinetic simulations.

It should also be possible to effectively suppress the numerical instability in the gyrokinetic case by increasing the parallel resolution and filtering grid-scale modes (method D in Ref. 4) even for first order splines the instability growth scales like k^4 (see Sec. III B), so a relatively weak filter should suffice. An additional issue we have not considered here but which has generally led to poor performance of electrostatic gyrokinetic PIC codes when electrons are treated kinetically is that the maximum time step possible also tends to decrease with system size, in order to resolve the (usually physically uninteresting) ω_H mode. Recent kinetic electron simulations have therefore largely been performed including electromagnetic codes, where this additional numerical issue is much less severe.

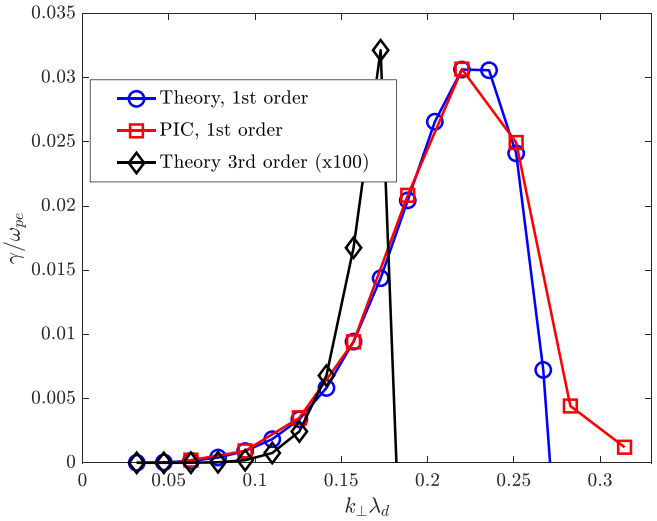

FIG. 2. Growth rate of the numerical PIC instability vs wavenumber, based on theory for first order splines (blue curve), PIC simulations using first order splines (red curve) and theory for third order splines (black curve, value scaled by a factor of 100 to make it visible on this graph). Numerical instabilities were too small to be detected in simulations with third order splines.

VI. CONCLUSION

"Resolving the Debye length" is a common prescription for avoiding numerical instability in standard PIC simulations, but standard PIC codes are often run in regimes where this condition is violated, and the analogous condition in gyrokinetic simulation cannot be met by refining the spatial grid. We have shown that the numerical instabilities in both standard and δf PIC codes, at least in the small time step limit, may be massively reduced by using smoother, higher order shape functions, and partly as a result, most modern PIC codes use higher order splines. There is, however, always a small residual numerical instability (see Appendix B), but for third order splines, the growth rates are more than one thousand times lower than the plasma frequency, and this is undetectable in massively over-resolved simulations carefully designed to examine this instability. It is generally a good idea to implement variable-order splines when building a PIC code (or at least two different orders of spline) in order to tune for maximum performance, and this allows a straightforward way to test whether these tiny remnant instabilities are relevant. In the gyrokinetic context, PIC codes typically use higher (normally at least second) order splines and energy conserving methods, and include electromagnetic effects[13,14] (which frustrate the instability by limiting the parallel wave speed to the Alfvén velocity, which is typically much lower than the electron thermal speed). In order to not just reduce the growth rate to undetectable levels but to remove the instability entirely, code authors should consider adding electromagnetic effects and using energy-conserving methods in new and existing codes.

APPENDIX A: DERIVATION OF THE NUMERICAL VLASOV EQUATION FOR THE DELTA-F PIC METHOD

As for physical plasmas, a statistical theory of the behavior of the numerical markers may be derived. For the standard PIC method, this leads to an equation for the modeled distribution f, which may be extended to consider finite-particle number effects.[9] We present an outline of how the numerical Vlasov equation associated with the δf method may be obtained. The steps are mostly completely analogous to the way the Vlasov equation is derived from many-body classical mechanics for physical plasmas. Essentially, this boils down to counting the number of markers entering and leaving a phase-space box, deriving a number density conservation equation, and using phase space conservation to express this in the form of Vlasov's equation.

We now follow the standard procedure for transforming particle equations to equations for distributions in the large marker-number limit, to show that a numerical equation still applies for δf PIC simulations. With the same trajectory equations as for standard PIC, and for a given field E, in the infinite-marker number limit, we have (for a phase space volume K and time interval τ giving an integration volume $C = K\tau$)

$$\int_C dV\frac{\partial f}{\partial t}=\sum_i\int_C dV\left[\frac{dw_i}{dt}+w_i\frac{dx_i}{dt}\frac{\partial}{\partial x_i}+w_i\frac{dv_i}{dt}\frac{\partial}{\partial v_i}\right]\delta(x-x_i)\delta(v-v_i),$$

after substituting in Eq. (13), where $dV = dxdvdt$. We then can use the chain rule to show $\partial_a f(a-b) = -\partial_b f(a-b)$, and since the delta function is zero except where $x = x_i$ and $v = v_i$, we have

$$\int_C dV\frac{\partial f}{\partial t}=\sum_i\int_C dV\left[-\Omega_i\frac{df_0}{dt}-w_i\frac{dx}{dt}\frac{\partial}{\partial x}-w_i\frac{dv}{dt}\frac{\partial}{\partial v}\right]\delta(x-x_i)\delta(v-v_i), \tag{A1}$$

where the equations of motion are now evaluated using the integration variables. At this point, we may move the sum into the integral (we are assuming uniform convergence) to find

$$\int_C dV\frac{\partial f}{\partial t}=-\int_C dV\frac{df_0}{dt}\sum_i[\Omega_i\delta(x-x_i)\delta(v-v_i)] - \int_C dV\left[\frac{dx}{dt}\frac{\partial}{\partial x}+\frac{dv}{dt}\frac{\partial}{\partial v}\right]\sum_i[w_i\delta(x-x_i)\delta(v-v_i)]. \tag{A2}$$

We then use integration by parts to rewrite the second integral as a divergence: because the equations of motion have a Hamiltonian character, the divergence of these zero. The second term may then be written using Gauss's law as the surface integral of a flux, and because a the number of markers entering or leaving this volume over a finite time-interval is large in the large-particle-number limit, the limit of this surface integral is well-posed. We then use the definition of δf in the large marker number limit to rewrite this term in a continuum form.

The first term on the RHS of Eq. (A2) involves the sum of Ω_i on the phase space volume: this is the sum of the nominal phase space volume associated with the relevant markers. Due to phase-space incompressibility, one can show that

$$\int_C dV\sum_i[\Omega_i\delta(x-x_i)\delta(v-v_i)]=\int_C QdV, \tag{A3}$$

in the large marker-number limit, with $Q = 1$ (the proof is by deriving a Vlasov equation for Q, which is initially unity and uniform, and is therefore always equal to 1). Phase-space conservation is a real requirement to ensure correct sampling, and hence accurate δf PIC simulations, and not just used to simplify the derivation.

Then, taking the region C to be small (i.e., a short time/phase space interval), we may assume df_0/dt, dv/dt, and dx/dt to be replaced by their averages on C, and taken out of the integral,

$$\int_C dV\frac{\partial f}{\partial t}=-\int_C dV\frac{df_0}{dt}+\frac{dx}{dt}\frac{\partial\delta f}{\partial x}-\frac{dv}{dt}\frac{\partial\delta f}{\partial v}. \tag{A4}$$

Using $f_0 = f - \delta f$, and evaluating the full derivative term-by-term, we then have

$$\int_C dV\frac{\partial f}{\partial t}=-\int_C dV\frac{dx}{dt}\frac{\partial f}{\partial x}-\frac{dv}{dt}\frac{\partial f}{\partial v}, \tag{A5}$$

with dv/dt evaluated, as for standard PIC, using the numerical field (see also the explanations in Aydemir[8]). Given an arbitrary but small integration domain C, we then have the standard Vlasov equation $df/dt = 0$. Note, as in standard kinetic theory, the self-consistent field depends on the particular configuration of markers in the simulation, so a complete derivation needs to consider $E_M = E + \tilde{E}$, where E is the continuum limit of the field, and $\tilde{E}$ is the fluctuating field that becomes small in the large marker-number limit. In a PIC simulation, $\tilde{E}$ is normally referred to as "noise," and in a physical plasma, it is this fluctuation that gives rise to collisions. The differing nature of discrete particle effects in PIC simulation and in physical plasmas has been explored to some extent elsewhere.[9]

APPENDIX B: ANALYTICAL ESTIMATES FOR THE PIC NUMERICAL INSTABILITY

The dispersion relation [Eq. (53) in Ref. 2, which we have reproduced for $N = 1$ as Eq. (15)] has the form

$$0=1+\frac{1}{k^2}\sum_q\frac{\sin\left(k_q\delta x/2\right)^{2N+2}}{\left(k_q\delta x/2\right)^{2N+2}}\int dv\frac{-\partial f_0(v)}{\partial v}\frac{1}{v-\omega/k_q}. \tag{B1}$$

Standard numerical techniques may be used to find the roots of this dispersion relation, but in some limits more direct estimates may be made.

We consider the limit $k \ll 1$ where the sum is dominated by the $q = 0$ component (this is the interesting limit for numerical instability as Landau damping stabilizes modes $k \gtrsim 1$). We then solve for the frequency at lowest order to find $\omega_0 = |\sin(k\delta x/2)/(k\delta x/2)|^{N+1}$. The imaginary part of the second term G may then be evaluated using contour integration in the $|\mathrm{Im}(\omega)| \ll 1$ limit as

$$G=i\pi\sum_{q\neq0}\mathrm{sign}(q+1/2)\frac{\omega_0^2}{k}\left(\frac{k}{k_q}\right)^{2N+2}\left.\frac{-\partial f_0(v)}{\partial v}\right|_{v=\omega_0/k_q}, \tag{B2}$$

and we have $\partial f_0/\partial v = -v\exp(-v^2/2)/\sqrt{2\pi}$.

This may directly be evaluated numerically, and the dominance of the terms with small $|q|$ means that few evaluations are required for reasonable accuracy. Using $k \ll 1$, we have

$$G=-i\sqrt{\frac{\pi}{2}}\frac{\omega_0^2}{k}\exp(-1/2k^2) + i\pi\omega_0^2k^{2N+2}\frac{\delta x}{2\pi}\sum_{q=1}^{\infty}\frac{\partial}{\partial v}\left(v^{2N+2}\frac{-\partial f_0(v)}{\partial v}\right)\Big|_{v=\omega_0\delta x/2\pi q}. \tag{B3}$$

The physical damping term dominates for sufficiently large k ($k \gtrsim 0.3$), but a numerical instability exists for small k with a peak at $k \sim 0.2$ for typical parameters. A good estimate of the growth rate may be made by keeping only the $q = 1$ term.

If we can assume $\omega_0 \sim 1$, the second term can be separated into a power law dependence on k, and a term dependent on δx with a narrow peak at $\delta x \sim 10$. Unfortunately, the peak growth rate of the numerical instability occurs where the real dispersion is very strongly affected by the shape function and $\omega_0 \ll 1$, so for most purposes a direct numerical evaluation is needed. Assuming $\omega_0 \sim 1$ yields a worst-case growth rate

$$G\leq-i\sqrt{\frac{\pi}{2}}\frac{1}{k}\exp(-1/2k^2)+ik^{2N+2}C_N, \tag{B4}$$

with $C_1 \sim 7$ and $C_3 \sim 500$. This worst-case estimate gives an incorrect estimate of order 1 for the maximum growth rate, but is useful for bounding the instability growth rate in limiting cases with a large grid spacing, or if a filter is used to remove the grid-scale modes: the scaling with k^{2N+2} means that even a fairly unrestrictive filter (for example, with a brick-wall low-pass Fourier filter) will be effective in significantly suppressing the instability.

For the purpose of clarifying the extent of the numerical instability, we may also consider the small δx limit where we have $k_q \gg 1$ for $q \neq 0$, and after some manipulation, we find

$$G = -i\sqrt{\frac{\pi}{2}}\left\{\frac{1}{k}\exp\left(-1/2k^2\right) - 4k^{2N+2}\left(\frac{\delta x}{2\pi}\right)^{2N+2}\sum_{q=1}^{\infty} q^{-(2N+2)}\right\}, \tag{B5}$$

and the growth rate of the numerical instability is $G/2i$. This expression for numerical growth is not particularly accurate for the worst-case scenario where $\delta x \sim 10$ and $k \sim 0.2$ but gives the correct scaling for smaller k and δx. The sum is dominated by the first term, and the total value is the Riemann zeta function evaluated with value $(2N + 2)$, which is almost exactly unity for $N \geq 1$ because the first term dominates.

REFERENCES

[1]C. Birdsall and A. Langdon, *Plasma Physics via Computer Simulation* (Taylor and Francis, New York, 1985).
[2]B. J. Sturdevant and S. E. Parker, J. Comput. Phys. **305**, 647 (2016).
[3]B. B. Godfrey and J.-L. Vay, J. Comput. Phys. **248**, 33 (2013).
[4]G. J. Wilkie and W. Dorland, Phys. Plasmas **23**, 052111 (2016).
[5]H. Lewis, J. Comput. Phys. **6**, 136 (1970).
[6]S. Markidis and G. Lapenta, J. Comput. Phys. **230**, 7037 (2011).
[7]S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T. Tran, B. McMillan, O. Sauter, K. Appert, Y. Idomura, and L. Villard, Comput. Phys. Commun. **177**, 409 (2007).
[8]A. Aydemir, Phys. Plasmas **1**, 822 (1994).
[9]A. B. Langdon, Phys. Fluids **22**, 163 (1979).
[10]T. D. Arber, K. Bennett, C. S. Brady, A. Lawrence-Douglas, M. G. Ramsay, N. J. Sircombe, P. Gillies, R. G. Evans, H. Schmitz, A. R. Bell *et al.*, Plasma Phys. Controlled Fusion **57**, 113001 (2015).
[11]T. Esirkepov, Comput. Phys. Commun. **135**, 144 (2001).
[12]J. Villasenor and O. Buneman, Comput. Phys. Commun. **69**, 306 (1992).
[13]A. Bottino, B. Scott, S. Brunner, B. McMillan, T. Tran, T. Vernay, L. Villard, S. Jolliet, R. Hatzky, and A. Peeters, IEEE Trans. Plasma Sci. **38**, 2129 (2010).
[14]E. Lanti, N. Ohana, N. Tronko, T. Hayward-Schneider, A. Bottino, B. McMillan, A. Mishchenko, A. Scheinberg, A. Biancalani, P. Angelino *et al.*, Comput. Phys. Commun. **251**, 107072 (2020).
[15]G. Wilkie, private communication (July 21, 2017).